

AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph beginning on page 30, line 20 as follows.

The customer workstation 20 is browser enabled and includes client applications responsible for presentation and front-end services. Its functions include providing a user interface to various MCI services and supporting communications with MCI's Intranet web server cluster 24. As illustrated in Figure 2, and more specifically described in the commonly owned[, co-pending] U.S. Patent 6,115,040 [Application _____ (D#11040)] entitled GRAPHICAL USER INTERFACE FOR WEB ENABLED APPLICATIONS, the contents and disclosure of which are incorporated by reference as if fully set forth herein, the client tier software is responsible for presentation services to the customer and generally includes a web browser 14 and additional object-oriented programs residing in the client workstation platform 20. The client software is generally organized into a component architecture with each component generally comprising a specific application, providing an area of functionality. The applications generally are integrated using a "backplane" services layer 12 which provides a set of services to the application objects which provide the front end business logic and manages their launch.

Please amend the paragraph beginning on page 31, line 30, as follows.

As will be described, each of the nMCI Internet suite of network management applications implements a set of common objects (CO) that minimizes the replication of code, and provides a framework in which a family of internet applications may be managed and created including: communications, I/O services to local resources, user authentication, internationalization, common look and feel, application management, and a model view controller (MVC) framework. The primary common object services for each of the suite of applications include: graphical user interface (GUI); application launch; window navigation among applications; inter-application communications; printing; user identity, session management, authentication, and entitlements; data import and export; logging and statistics; error handling; version management; and messaging services. The use of a set of common objects for implementing the various functions provided by the integrated interface system of the present invention, and particularly the use of browser based objects to launch applications and pass data therebetween is more fully described in the above referenced, commonly owned U.S. Patent 6,115,040 [copending application _____ (D#11040)] GRAPHICAL USER INTERFACE FOR WEB ENABLED APPLICATIONS, and Appendix A, attached to that application, provides descriptions for the common objects which includes various classes and interfaces with their properties and methods.

Please amend the paragraph beginning on page 45, line 9, as follows.

An illustrative example of the nMCI Interact logon Web page may be found in [co-pending] commonly owned U.S. Patent [Application] No. 6,115,040 [____ [D#11040]] which typically includes a name field and a password field for the user to enter. After the user is properly authenticated via the logon page, the nMCI Interact home page is retrieved.

Please amend the paragraph beginning on page 56, line 15, as follows.

With regard to user selection of the select enterprise menu option or toolbar button in Figure 16, the browser displays the web page having a dialog box as shown in commonly owned[, co-pending] U.S. Patent 6,587,836 [application No. _____ (D#11042)] entitled AUTHENTICATION AND ENTITLEMENTS FOR USERS OF WEB BASED DATA MANAGEMENT PROGRAMS, the contents and disclosure of which are incorporated by reference as if fully set forth herein, which enables an administrator to work with a different enterprise, as well as add an enterprise to their enterprise list and additionally includes the ability to set up new users or modify various options available to existing users.

Please amend the paragraph beginning on page 56, line 29, as follows.

During the StarOE add or modify procedure described above, security information regarding customer entitlements for application services may also be initialized as described in commonly owned[, co-pending] U.S. Patent 6,587,836 [application No. _____ (D#11042)] . For example, a screen may be presented for setting up Toll Free Network Manager ("TFNM") security information and is displayed when TFNM is ordered or modified. Preferably, a user's TFNM security profile includes at least one corp id, with each corp id having an associated racf id. Preferably, a setup security object handles the process of setting up security for each application. A constructor for this object initializes the user Id and a modify flag as passed in from the StarOE client application 154. The object retrieves the toll free hierarchy from the StarOE server 39 using the "get hierarchy" message. The client application 154 sends the enterprise Id, and toll-free flag in the request, and the StarOE server 39 returns the list of toll-free corp ids for the enterprise. If the modify flag is set, a "get security" message is sent to the server 39 to retrieve the user's TFNM security profile. As a displayed tree structure is loaded with each toll free corp id, racf id is entered by a user. When the submit button is pressed, the setup security object calls its send security method which causes the formatting and sending of "setTFNM security" message to the StarOE server 39. When the StarOE server 39 receives the message, it sets the security accordingly for the TFNM application.

Please amend the paragraph beginning on page 65, line 9, as follows.

As further described in the herein incorporated[, co-pending] U.S. Patent 6,587,836 [application No. _____ (D#11042)], the StarOE server 39 is preferably implemented utilizing object oriented programming (OOP). As an example, when a "get hierarchy list" message is initiated at the client application to invoke retrieval of a toll free corp Id list from the server 39, a "Hierarchy" class may be instantiated which includes a Get() method to determine which Hierarchy product is to be retrieved (e.g., Toll free, Vnet/CVNS, or Vision) and to return the appropriate information. Another object may be invoked to format the data into a response message and return the message back to the client. As another example, when a "get application list" request message is initiated at the client application, an "Application" class may be instantiated which encapsulates the interface into a database table (not shown) having applications information. Particularly, the Get() method in this class accesses the Applications table in the database and return the list of application codes and their descriptions. The details of the message format, including request and response messages, are described in commonly owned[, co-pending] U.S. Patent 6,587,836 [application No. _____ (D#11042)].

Please amend the paragraph beginning on page 68, line 6, as follows.

Additional authentication and entitlement data may be transmitted from a corporate order entry system ("CORE") 292 which generates two sets of hierarchy files on a daily basis. One set comprises deltas only, the other comprises a full hierarchy. Notification is made to the StarOE when these are available. As described in co-pending U.S. Patent 6,587,836 [Application No. _____ (D#11042)], StarOE performs a reconciliation process to update the hierarchy files.

Please amend the paragraph beginning on page 69, line 16, as follows.

As mentioned herein, and in greater detail in commonly owned, co-pending U.S. Patent 6,631,402 [Application No. _____ (D#11050)] entitled INTEGRATED PROXY INTERFACE FOR WEB BASED REPORT REQUESTOR TOOL SET, the contents and disclosure of which is incorporated by reference as if fully set forth herein, the data architecture component of the networkMCI Interact system focuses on the presentation of real time (Un-priced) call detail data and reports, such as presently provided by MCI's TrafficView ("TVS"). Server, and priced call detail data and reports, such as presently provided by MCI's operational data store "StarODS" Server.

Please amend the paragraph beginning on page 71, line 1, as follows.

The Report Manager (*RM*) server 250 is an application responsible for the synchronization of report inventory with the back-end "Fulfilling" StarODS server 400 and Traffic View server 500; retrieval

of entitlements, i.e., a user's security profiles, and report pick list information, i.e., data for user report customization options, from the StarOE server 39; the transmission of report responses or messages to the Dispatcher server 26; the maintenance of the reporting databases; and, the management of metadata used for displaying reports. In the preferred embodiment, the RN server 250 employs a Unix daemon that passively listens for connect requests from the GUI client applications and other back-end servers and deploys the TCP/IP protocol to receive and route requests and their responses. Particularly, Unix stream sockets using the TCP/IP protocol suite are deployed to listen for client connections on a well-known port number on the designated host machine. Client application processes, e.g., report requestor 212, desiring to submit report requests connect to RN 250 via the dispatcher 26 by providing the port number and host name associated with RN 250 in a request message. Request messages received by the RN server are translated into a "metadata" format and validated by a parser object built into a report manager proxy 250 that services requests that arrive from the GUI front-end. If the errors are found in the metadata input, the RN 250 will return an error message to the requesting client. If the metadata passes the validation tests, the request type will be determined and data will be retrieved by the fulfilling server in accordance with the meta data request after which a standard response is sent back to the requesting client. As shown in Figure 10, interface sockets 252 are shown connecting the Dispatcher server 26 and the RN server 250 and, other socket connections 254, 256 are shown interfacing with respective back end servers 400 and 500. In one embodiment, as described in commonly owned, co-pending U.S. Patent Application No. 09/159,404 [_____] (D #11567) entitled INTEGRATED PROXY INTERFACE FOR WEB BASED DATA MANAGEMENT REPORTS, the contents and disclosure of which is incorporated by reference as if fully set forth herein, a back-end midrange application known as the TrafficView System receives the metadata requests to provide unpriced traffic call detail and reporting data through messaging interface 256 to the Report Manager. Additionally, as described in commonly owned, co-pending U.S. Patent 6,377,993 [Application No. _____] (D #11044) entitled INTEGRATED PROXY INTERFACE FOR WEB BASED DATA MANAGEMENT REPORTS, the contents and disclosure of which is incorporated by reference as if fully set forth herein, a back-end midrange application known as the StarODS server receives report requests for priced call detail data through a Talarian smart socket messaging interface 350 to the Report Manager. Additionally, as shown in Figure 10, the priced and unpriced data is FTP'd directly to the Inbox Server and a notification message is sent to the report manager server 250 from the Traffic View server 500. Although not shown in Figure 10, it should be understood that the RM 250 server can manage reporting data for customer presentation from other back-end and legacy servers including, e.g., Event Monitor and Service Inquiry servers, etc., in order to present to a customer these types of network management and reporting data.

Please amend the paragraph beginning on page 90, line 20, as follows.

As described in above-referenced[, co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)], and particularly Appendices A-G provided therein, the following types of metadata requests and responses that may be generated by the StarWRS Report Requestor 212 and Report Manager 250 components include: 1) Get/Send report template list (GRTL/SRTL) - which request enables retrieval of the list of all standard report templates for all products and is used only to obtain general report information, e.g., report title, description, etc.; 2) Get/Send report template detail (GRTD/SRTD) - which request retrieves the details of a specific standard report template; 3) Get/Send user report list (GURL/SURL) - which request retrieves the list of all user reports for the report format selected from a user report table and is used only as a request for general report information, e.g., report title, status, etc.; 4) Get/Send user report detail (GURD/SURD) - which request retrieves the details of a specific user's report; 5) Add report definition/Acknowledgment (ARD/ARDA) - which requests addition of a user-created report to a user report table. If the report is a scheduled report, this request is also communicated to the fulfilling server at the time the report is due; 6) Delete report definition/Acknowledgment (DRD/DRDA) - which request deletes a user-created report from the user table; 7) Copy report definition/Acknowledgment (CRD/CRDA) - which request creates a duplication of the report the user is editing (other than the report title) and creates a new report ID for it; 8) Update Reporting Schedule/Acknowledgment (URD/URDA) - which request updates the scheduling information on a report without having to send a Delete and Add request; and, 9) Get Pick List/Acknowledgment (GPL) - which request enables the Report Requestor 212 to get a pick list provided by StarOE server.

Please amend the paragraph beginning on page 93, line 24, as follows.

The process for generating a report for StarODS priced call detail data is described in detail in aforementioned [co-pending] US. Patent 6,377,993 [Application serial No. _____ (D#11044)], and, for TVS unpriced call detail data, in aforementioned co-pending U.S. Patent Application [serial] No. 09/159,404 [(D#11567)]. Generally, whether the report is to be currently run for immediate ad hoc reporting, or, is scheduled for normal scheduled reporting, the following sequence of operations, as indicated at steps 370-395, Figures 11(b) - 11(c), are performed: First, in response to receipt of the ARD message, e.g., submitted to the fulfilling server by the Report Scheduler, the fulfilling server completes the report and compresses the report/data, as indicated at step 370. Then, the report/data is "pushed", implementing FTP, to the fulfilling server's directory on the Inbox server 270, as indicated at step 373. Each application server, e.g., TVS server 550 (Figure 10), is responsible for generating unique file names within their directory on the Inbox server 270. For example, the following directory and file naming conventions used for reports generated by the TrafficView server are labeled inbox\files\tvs with text files

having the suffix *.txt or *.txt.zip (compressed), and comma separated files having a suffix *.csv or *.csv_zip (compressed). The fulfilling server then verifies that the FTP process was successful, as indicated at step 376, and, at step 379, a notification is sent by the fulfilling server to the Report Manager to notify the Report Manager server 250 of the location of a scheduled report. This is accomplished by using a "NRL" metadata message.

Please amend the paragraph beginning on page 95, line 1, as follows.

Aforementioned Appendix B of [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)] provides a table comprising the Notify Report Location parameters used for the NRL Metadata messaging sent by a fulfilling server to the RN Server 250 when a requested report is complete. Also provided in above referenced Appendix B is the acknowledgment table sent back to the fulfilling server in response. An example NRL message sent from the TVS server 500 to the RN server 250 can be found in commonly owned, co-pending U.S. Patent Application No. 09/159,404 [_____ (D#11567)].

Please amend the paragraph beginning on page 96, line 14, as follows.

Above referenced Appendix F of [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)] details the parameters that are passed in the GET METADATA messaging for indicating to the Report Viewer how to display a requested report. An example message in metadata format to initiate the generation of a MTD file corresponding to a user-created report for StarQJS priced call detail data and TVS unpriced call detail data may be found in [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)].

Please amend the paragraph beginning on page 97, line 6, as follows.

Above referenced Appendix C of [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)] provides a table showing the fields for the metadata messaging between the RN server 250 and the Inbox server 270 for adding an item into the StarWRS system Inbox server 270, and the respective acknowledgment message format back from the Inbox server. In the add "A" message found in Appendix C, the "LOC" field includes information about where the data is located. An example metadata message indicating to the Inbox server that an unpriced TVS fulfilling server report is available is described in [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)]. Particularly, the RN server supplies a metadata "A" message to the Inbox indicating the FTP file location. Via the report viewer, the report is now available for viewing,

downloading, saving, or printing by the user, as indicated at step 395, and as described in further detail in [co-pending] commonly owned U.S. Patent 6,385,644 [Application Serial No. _____ (D#11041)].

Please amend the paragraph beginning on page 98, line 19, as follows.

Referring back to Figure 10, the Report Viewer 215 interfaces with the user's Inbox 210 for presenting to the customer the various type of reports received at the Inbox. Preferably, all Report Requestor and Report Viewer applications communicate with the RN server 250 through the use of the common object communication classes, as described in greater detail in commonly-owned[, co-pending] U.S. Patent 6,385,644 [Application Serial No. _____ (Atty. Dckt. 11041)] entitled **MULTI-THREADEDWEB-BASED USER INBOX FOR REPORT MANAGEMENT**, the contents and disclosure of which is incorporated by reference as if fully described herein.

Please amend the paragraph beginning on page 100, line 26, as follows.

Particularly, as shown in Figure 16, the Inbox server 270 interface with the Inbox Client 210 supports messaging that enables the User to remove an item from the Inbox, e.g., delete a report, or, to delete all items from the Inbox, e.g., for a particular Enterprise and User ID as well as other associated reports. Above referenced Appendix G of [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)] illustrates the parameters used in the metadata messaging between the Inbox client and the Inbox server. Particularly, the List "L" message is a synchronous request for a list of all Inbox items for a specific user. The Inbox fetch "F" function is a bulk transfer request that enables bulk transfer of the requested file to the Inbox client.

Please amend the paragraph beginning on page 102, line 16, as follows.

As described in greater detail in [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)] the Report Scheduler server 260 is additionally capable of updating the User report status table and, preferably, is provided with a tracking mechanism for tracking the scheduling of user reports. If the report is an Ad hoc report, it is marked as inactive in the user report table once the status is complete.

Please amend the paragraph beginning on page 103, line 4, as follows.

In Figure 14(a), there is shown the high-level logical approach of the StarODS data management system 400 integrated with the StarWRS component 200 of the nMCI Interact architecture. Generally, the data management system 400 of the invention, referred to herein as "StarODS", provides customers with priced reporting data pertaining to telecommunications services. Although the description herein

pertains to priced billing data, it should be understood that the principles described herein could apply to any type of reporting data. Through StarWRS web-based reporting, the StarODS system provides priced reporting data and implements a DataMart approach for maintaining the data used for customer reporting. StarODS stores and incrementally processes customer's priced data included in call detail records, and loads this processed data in Data Marts in a manner such as described in commonly owned [, co-pending] U.S. Patent 6,714,979 [Application No. _____ (D#11568)] entitled DATA WAREHOUSING INFRASTRUCTURE FOR WEB-BASED REPORTING TOOL, the contents and disclosure of which are incorporated by reference as if fully set forth herein. From these data marts customer's priced reporting data may be provided to customers on a daily basis via the StarWRS reporting system.

Please amend the paragraph beginning on page 110, line 23, as follows.

Above-referenced commonly owned U.S. Patent 6,377,993 [Application No. _____ (D#11044)] describes in greater detail the application programming interface "API" whereby the RN server 250 publishes the message to the Decision Support Server in response to its receipt of a report request. Similarly, a DSS/Inbox API is provided to manage FTP transmission of completed customer report files including: error handling, retry logic, and the ability to maintain the file name and location of where report files are stored. Particularly, the DSS/Inbox API sends the report file to the Inbox (Figure 14 (a)).

Please amend the paragraph beginning on page 113, line 1, as follows.

In Figure 14(b), a Talarian receiver, referred to herein as a Talarian Interface Object ("Tb") 350, is a process that receives the Talarian message, manages the GMD functionality, and posts updates to the request table 493 and request status table 494. Appendix "I" of above-referenced, [co-pending] commonly owned U.S. Patent 6,377,993 [Application No. _____ (D#11044)] illustrates the contents of the ODS Request table 493 which is the table maintained for the purpose of holding specific report request information from the received ARD message, and, a Request Status table 494, for tracking the status of IJSS processes for the current request. As further shown in Figure 14(b), the receiver 350 inserts the message received from an arbitrator into the request table 493 and request status table 494 along with the priority, timestamp and status fields. The request status table resides on the ODS database and the messages are stored in the queue to provide queuing, log and tolerance from the failures. To determine the pending messages to be processed, a status field and history_stat flags are used.

Please amend the paragraph beginning on page 119, line 5, as follows.

More particularly, as described in further detail in [co-pending] commonly owned U.S. Patent 6,377,993 [Application No. _____ (D#11044)], the report result set from Decision Suite is input to a Formatter module which performs various report result transformations including: 1) Converting of column headers generated by Information Advantage into appropriate column ids that are recognizable to the StarWRS client viewer functionality (as indicated at step 650); 2) Provide subtotalling for specific requested "subtotal by" columns in the format required by the StarWRS client interface (as indicated at step 653) and provides report-based totals as requested by customer; 3) converting binary stream data file to ASCII text file (as indicated at step 655); 4) implementing Replacelogic, e.g., replacement of "TAB" delimiters with appropriate "Comma" field delimiters (as indicated at step 657); 5) implementing Repeat/Padding logic, i.e., identifying compressed columns/values and decompressing (or repeating) the values that were compressed; 6) providing alphanumeric translations for any encoded data elements returned in the result set data file (as indicated at step 659); and, 7) adding new computed/derived columns, e.g., percents, averages of column data values, etc., as requested by customers on specific reports.

Please amend the paragraph beginning on page 132, line 17, as follows.

The statistics that are gathered for each subscriber's toll-free number in the TVS system of the invention include: total completions, total call duration, total attempts, total switch control call, total Network Control System (NCS) blocked, total NCS rejected, total network blocked (all routes busy), total supp code blocked, and out-of-band blocked. Appendix I of co-pending U.S. Patent Application No. 09/159,404 [_____] (D#11567)], provides a summary table processing algorithm detailing the collection of statistics by the GSE and the TVS summary table processing.

Please amend the paragraph beginning on page 133, line 22, as follows.

Appendix I of co-pending U.S. Patent Application No. 09/159,404 [_____] (D#11567)] depicts the algorithms implemented in TVS stats_counter process 570 for generating statistics data tables so that TCR records may be processed in batches. As shown, the processes include: a summary table process which process generates statistics for call summary data; a NPA table process; Country table process and Termination table process. The stats_counter 570 enables multiple processes to be run at the same time on the same machine. To allow an arbitrary number of Stats_Counter processes, the stats databases are organized as a series of configurable tables, e.g., "C_Tables" 572, which are temporary tables that the stats counters first insert records to. These tables are identical to normal statistics tables with the exception that they include a field for the date in them. In accordance with the provision of C_tables, a

pending_stats_list table and stats_table_usage_list table are used to keep track of what data is in the C_tables, and to drive the movement of data from the C_tables to a more permanent database tables 574.

Please amend the paragraph beginning on page 135, line 10, as follows.

Table 1 of co-pending U.S. Patent Application No. 09/159,404 [_____] (D#11567)), depicts an example pending_stats_list table which comprises a directory of what the stats_counter is working on, or finished with. Each record represents a name of a c_table that contains statistics, and dates that are contained in this c_table. The report generator process, and on-line access use this table to determine if there is any data in the c_tables that they may be interested in, and what the table name is. The Stats_counter processes insert records into this table, and data_mover processes 573, shown in Figure 21, remove entries from this table.

Please amend the paragraph beginning on page 135, line 23, as follows.

Table 2 of co-pending U.S. Patent Application No. 09/159,404 [_____] (D#11567)), depicts an example stats_table_usage_list table which comprises a list of all the c_tables that are configured and used by the stats_counter processes and data_mover processes to allocate tables amongst themselves. The number of records in this table remains static. Stats_counter processes 570 update the "used_by_process" field with their process name when they are in control of that table. At the top of the hour, they may change the used_by_process to "MOVER", and attempt to find another table that is unallocated. The movers change the used_by_process name to "NONE" when they have completed moving data from that c_table. In the preferred embodiment, there are four types of movers are currently configured to run: NPA, summary, country, and termination. Each type of mover looks in the pending_stats_list for the name of the c_table of the same type with a usage_flag of "2", for instance, and the earliest date. The mover then transfers the data for this date from the "c_table" to appropriate the permanent table. When the data transfer is finished, the matching record in pending_stats_list is deleted. If there are no more entries for this "c_table" in pending_stats_list, the mover process takes the precautionary step of searching the "c-table" for additional data that was not noted in pending_stats_list. Entries are then added to pending_stats_list for any data found in the "c_table". If no additional data is found, used_by_process in stats_table_usage_list is changed from "MOVER" to "NONE" for this "c_table".

Please amend the paragraph beginning on page 139, line 7, as follows.

As described herein, when the user requests call detail for a particular period of time, this request is translated by the StarWRS component into a metadata file which is sent to TVS in the manner

described herein. Users schedule reports for execution using the Report Scheduler in StarWRS in the manner as described herein and in [co-pending] commonly owned U.S. Patent 6,631,402 [Application No. _____ (D#11050)]. When the user has completed report selection, modifications and scheduling, the StarWRS Report Scheduler component 260 creates a metadata message comprising this information which file is passed to TVS in real time. The TVS then uses this file to formulate a query and runs the report for the scheduled time period.

Please amend the paragraph beginning on page 141, line 27, as follows.

From an RTM window provided on a web-screen page, the customer is enabled to select or modify his/her predefined RTM use profile, as indicated at step 740, Figure 23(a). Specifically, the customer may create or edit their user profile, for example, by entering selection criteria such as: 800/8xx or Vnet number to report, a polling interval, and time zone. The user may also delete a user profile. The entered selection criteria may be saved by the subscriber as a new user profile for storage in the TVS server Level_of_use tables, as indicated at steps 744 and 745, or submitted directly to the TVS server, as indicated at steps 749 and 750. It should be understood that all TVS RTM functionality as described in [co-pending] commonly owned U.S. Patent 5,825,769 [Application No. 08/587,381 filed January 17, 1996], entitled SYSTEM AND METHOD THEREFOR OF VIEWING IN REAL TIME CALL TRAFFIC OF A TELECOMMUNICATIONS NETWORK, the contents and disclosure of which is incorporated by reference as if fully set forth herein, may be available to the customer.

Please amend the paragraph beginning on page 144, line 17, as follows.

In view of the foregoing, a subscriber via a client workstation running a Web browser can monitor in real time, or, in addition, using the RTM system, has the ability to monitor in substantially real time, the operation of the network as it relates to the calls directed to that subscriber's special service call number(s). For example, the subscriber may see in real time how many calls are being attempted minute by minute, how many calls are being allowed through the network, how many calls are incompletes, how many calls are blocked, etc. This ability to monitor the operation of the network gives the subscriber the ability to decide in real time the specific actions that need to be taken. For instance, if there is an abnormal number of incompletes for a given period, the subscriber can look at the specific call records that made up those incomplete calls. In the manner described herein and in co-pending U.S. Patent Application No. ____ (D#11046), the subscriber may then request the management of the network to restructure the network so as to reroute the incoming calls of the subscriber to different locations where they may better be handled.

Please amend the paragraph beginning at line 147, line 12, as follows.

Figure 25(a) illustrates the high-level design of the Service Inquiry application 2200 including the client application 2250 and server 2300 components. As shown, Service Inquiry requires integration with a number of external systems and utilizes the Common Objects Framework for inter-application communications. Interfacing with the Service Inquiry application server 36 via the common objects framework are the StarOE server, e.g., for user profile information, as well as other Service Inquiry specific data, and, the CSM legacy host that provides the ability to query, status, and take action on service inquiries. Communication between the SI application server 36 and CSM 40(a) is via Registry middleware, such as described in commonly owned[, co- pending] U.S. Patent 5,790,809, [Application No. 08/560,550] incorporated by reference herein. Figure 3 shows COF-based inter-application communication between Service Inquiry and StarOE. It should be understood that if an external system does not use the COF. Service Inquiry may utilize that system's API set and communication mechanism for inter-application communication. The above-referenced Registry system has a number of options for inter-application communication, including both Java and CORBA interfaces.

Please amend the paragraph beginning on page 148, line 27, as follows.

As described herein and in greater detail in [co-pending] commonly owned U.S. Patent 6,859,783 [Application No. _____ (D#11673)] entitled INTEGRATED INTERFACE FOR WEB BASED CUSTOMER CARE AND TROUBLE MANAGEMENT, the contents and disclosure of which is incorporated by reference as if fully set forth herein, the SI application server 2300 interfaces with the Legacy Backend 40(a), CSM/SI through a Requester object 2310 and Receiver object 2350 as shown in Figure 25(b). Particularly, the SvcInqCSMRequester object 2310 is the class that represents the requester which takes the request data that comes from the Front-End/Client application through the Transaction Manager 2320, builds the CSM/SI request transactions by interacting with the Translator classes 2380 and ships off the requests to CSM. The request data that comes from the Front End/Client is an array of strings that are required from the customer for the request to be made. Minimal information is passed from the client to reduce the communication overhead from the client to the SI application server. All other information is packaged in the Requester. Particularly, the Requester object 2310 uses the SvcInqRegistryHeader and SvcInqSIHeader classes in the Translator 2380 to build the "Registry Header" and "SI Header" strings that are required for the CSM/SI request transactions. It also talks to the SvcInqActivity or the SvcInqRemarks classes to build the data portion of the CSM/SI requests. Once the CSM/SI Transaction String is formatted the actual request to CSM is made. Sending the transaction to CSM's Standard Interface (SI) via Registry classes does this.

Please amend the paragraph beginning on page 168, line 16, as follows.

Questions are the main component in a QuestionTree. A Question has a vector of group identifiers that indicate the groups to which it belongs. A Question has a vector of RegistryEntry instances 2640a called choices. When the user "answers" the Question, the answer is set to the selected choice; i.e., the selected RegistryEntry. Short answer or text answer questions are a specialization of this behavior. Within each group of Questions, there is one question that is designated as the decision point which is used to determine the next group of Questions that need to be presented to the user. As a Registry Entry may contain a nextGroupID, the nextGroupID of the RegistryEntry instance selected as an answer to a decision point Question is used to derive the next group of Questions. Occasionally, the only difference between the two groups of Questions is the inclusion or exclusion of a particular Question. One solution is to create two identical groups, one with the optional question and one without and rely on the decision point mechanism. In the preferred embodiment, an optional parent-child relationship between Questions is created. The inclusion/exclusion of a Question (child) in a group is based on the answer to a previous Question (parent). A child Question maintains a reference to one of the possible choices (RegistryEntry) of the parent Question. If the parent Question's answer is the same as the child Question's parentAnswer, the child Question is included in the group; otherwise, it is excluded from the group. Details regarding the process by which a system administrator may generate trouble ticket queries corresponding to particular types of trouble tickets are provided in above referenced, [co-pending] commonly owned U.S. Patent 6,859,783 [Application No. _____ (D#11673)] entitled INTEGRATED INTERFACE FOR WEB BASED CUSTOMER CARE AND TROUBLE MANAGEMENT.

Please amend the paragraph beginning on page 176, line 1, as follows.

Once the customer has logged into TFNM and has received the StarOE security message, a communication is made from the TFNM server 840 to NetCap 290 requesting a user security profile. Particularly, the messaging system implemented for all communications between the TFNM server and NetCap is referred to herein as "Registry", such as shown and described in commonly-owned[, co-pending] U.S. Patent 5,790,809 [Application No. 08/560,550], the contents and disclosure of which are incorporated by reference as if fully set forth herein. Security from NetCap is by Racf Id and Corp Id. For each Corp Id a user has access to, that user must have a Racf Id. If a user has Enterprise level security, then the list of Corps under that Enterprise within NetCap have the same security as the Enterprise. Particularly, in response to a user login, in the preferred embodiment, a TFNM server application is executed. From this application, the TFNM server instantiates a Profile Manager Java object which is registered with CORNI and called upon to invoke further objects relating to the following

user profile, e.g. preferences, user security profiles, i.e., for tracking customer entitlements/privileges including rights for creating or modifying specific TFNM routing plans or generating QUIK or IMPL orders; and, session management, i.e., objects which encapsulate the state and behavior associated with a specific user login, e.g., time logged in.

Please amend the paragraph beginning on page 178, line 4, as follows.

In the preferred embodiment, as shown in Figure 26, the TFNM server 840 communicates a plan/data sync message 843 via Registry messaging to NetCap. Appendix A of commonly owned[, co-pending] U.S. Patent 6,574,661, [Application No. _____ (D11046)] entitled INTEGRATED PROXY INTERFACE FOR WEB BASED TELECOMMUNICATION TOLL-FREE NETWORK MANAGEMENT, the contents and disclosure of which is incorporated by reference herein, illustrates the Registry message call "NPSNC" which is the request to sync a plan and transmitted from the TFNM server to NetCap. A variety of Registry response messages for this request is provided in Appendix B of above-referenced[, co-pending] commonly owned U.S. Patent 6,574,661, [Application No. _____ (D11046)].

Please amend the paragraph beginning on page 187, line 20.

Appendix A of above referenced[, co-pending] commonly owned U.S. Patent 6,574,661, [Application No. _____ (D11046)], illustrates the Registry message calls that are transmitted from the TFNM server to NetCap for the IMPL/TEMP IMPL order and the corresponding NetCap responses. Included is the message for submitting an IMPL order (NIMPL) to NetCap.

Please amend the paragraph beginning on page 188, line 17, as follows.

Appendix B of above-mentioned[, co-pending] commonly owned U.S. Patent 6,574,661, [Application No. _____ (D11046)] illustrates the Registry message calls that are transmitted to the TFNM server from NetCap in response to the submitted IMPL order. Included is the message indicating successful processing of the IMPL request (NSUCS) and the message indicating completion of the order in NetCap (UCOMP). The TFNM server passes this information on to the user via CORMI messaging over the IITTPS connection. If the user is still logged on, this acknowledgment appears as a pop-up message on their screen, as indicated via line 826 in Figure 26. If the user has logged off, TFNM retains the acknowledgment that the IMPL has been received and saved for the next user login. Likewise, when an IMPL has been transmitted to NetCap and either implemented or terminated, NetCap sends a registry message back to the TFNM server which, in turn, passes this information back to the user via HTTPS connectivity.

Please amend the paragraph beginning on page 193, line 10, as follows.

The above referenced Appendix A of [co-pending] commonly owned U.S. Patent 6,574,661 [Application No. _____ (D11046)] also illustrates the Registry message calls that are transmitted from the TFNM server to NetCap for the QUIK/TEMP QUIK order and the corresponding NetCap responses. Included as the message for submitting an QUIK order (NQQUIK) to NetCap.

Please amend the paragraph beginning on page 194, line 1, as follows.

Appendix B of [co-pending] commonly owned U.S. Patent 6,574,661 [Application No. _____ (D11046)] also illustrates the Registry message calls that are transmitted to the TFNM server from NetCap in response to the submitted QUIK order. Included is the message indicating successful processing of the QUIK request (NSUCS) and the message indicating completion of the order in NetCap (UCOMP). The TFNM server then passes this information on to the user via CORMI messaging over the HTTPS connection. If the user is still logged on, this acknowledgment appears as a pop-up message on their screen, as indicated via line 826 in Figure 26. If the user has logged off, TFNM retains the acknowledgment that the QUIK order has been received and saved for the next user logon. Likewise, when a QUIK has been transmitted to Netcap and either implemented or terminated, NetCap sends a registry message back to the TFNM server which, in turn, passes this information back to the user via HTTPS connectivity.

Please amend the paragraph beginning on page 196, line 8 as follows.

Appendix A of [co-pending] commonly owned U.S. Patent 6,574,661 [Application No. _____ (D11046)] also illustrates the Registry message calls that are transmitted from the TFNM server to NetCap for un-approving an order (NOUAP), zapping an order (NOZAP), and, requesting pending order data (NPIUO). Corresponding NetCap responses are provided in Appendix B of [co-pending] commonly owned U.S. Patent 6,574,661 [Application No. _____ (D11046)].

Please amend the paragraph beginning on page 249, line 5, as follows.

An exemplary Broadband web-page BB Main Display screen 1720 is shown in Figure 34(a) which presents a variety of user-selectable Broadband options including: 1) an Inbox option 1722 enabling a user to retrieve their Broadband reports; 2) an option 1724 enabling a user to view SNMP alarms; 3) an option 1726 enabling a user to specify reports to be suppressed; 4) an option 1728 enabling a user to retrieve Broadband reports that have been archived; 5) an option 1730 enabling a user to receive information regarding their circuit locations, 6) an option 1732 enabling the "get/set" functionality for

providing meaningful labels to circuit locations to improve report readability; and, 7) an option 1734 enabling the retrieval of a map viewer application for generating maps portraying the customer's virtual networks. Details of each of these Broadband applications may be found in commonly-owned[, co-pending] U.S. Patent 6,574,661 [Application No. _____ (D11046)] entitled INTEGRATED PROXY INTERFACE FOR WEB BASED BROADBAND TELECOMMUNICATIONS MANAGEMENT, the contents and disclosure of which is incorporated by reference herein.

Please amend the paragraph beginning on page 257, line 5, as follows.

Furthermore, as shown in Figure 34(a), the selection of the archive option 1728 will enable the presentation of a BB view archive report screen 1760 an example screen of which is shown in Figure 34(d). From the screen of Figure 34 (d), the user is enabled to select the archived report to be displayed at the BB Inbox by name, type, scheduling period, and reporting time period in entry fields 1769. Details of the process flow in generating the Broadband archive reports corresponding to the archive option screen of Figure 34(d) is described in above-referenced, [co-pending] commonly owned U.S. Patent 6,490,620 [Application No. (D#11048)].

Please amend the paragraph beginning on page 263, line 2, as follows.

Thus, as further shown in Figure 34(b), the selection of the SNMP Alarm option 1724 enables the presentation of a BB alarm panel screen 1765 an example screen of which is shown in Figure 34(f). From the screen of Figure 34(f), the user is enabled to retrieve and view their virtual network alarm conditions including the following indications: an alarm type; circuit id; alias; alarm severity level; alarm trap level; alarm description; and, date of the alarm. Details regarding the process flow in providing near-real time Broadband alarms is found in above-referenced, [co-pending] commonly owned U.S. Patent 6,490,620 [Application No. (D#11048)].

Please amend the paragraph beginning on page 267, line 3, as follows.

Above-referenced, [co-pending] commonly owned U.S. Patent 6,490,620 [Application No. (D#11048)] describes the valid customer attributes that a client may get.

Please amend the paragraph beginning on page 298, line 3, as follows.

As described above, the client webstation 1130 provides a web-based graphical user interface (GUI) offering data management and data presentation features for the call manager system. The web-based front-end GUI is typically written using the Java programming language to insure platform independence. The client webstation 1130 typically includes a web browser with Java applets for the

interface for providing access to the call manager webstation application from a standard web browser, e.g., Internet Explorer V4.01. In addition, the networkMCI Interact common objects, described in the above-referenced, [co-pending] commonly owned U.S. Patent 6,115,040 [Application Serial No. _____ (D#11040)] and the contents and disclosure of which are incorporated by reference as if fully set forth herein, are used for implementing many functions needed for client/server communications protocols. The Java applets generally reside on the web servers 1132 and are dynamically downloaded to the client browsers (client webstations) 1130 when the Uniform Resource Locator (URL) for the call manager webstation client GUI application is accessed.

Please amend the paragraph beginning on page 319, line 3, as follows.

Above-mentioned [co-pending] commonly owned U.S. Patent 6,611,498 [Application No. _____ (D#11053)] illustrates a class diagram for a call manager global attribute display icon. A scenario diagram illustrating the class interactions when setting the global attribute and a scenario diagram showing the class interactions when the call manager application is launched from a web home page having the backplane applet is also described in [co-pending] commonly owned U.S. Patent 6,611,498 [Application No. _____ (D#11053)]. Upon the call manager application launch and initialization, the main toolbar sets its icon. For example, the browser starts the backplane applet which launches the CMAApp by calling its init method. The CMAApp calls setIcon method for setting up an icon for the main toolbar. The main toolbar may be implemented using a view of a model in a MVC paradigm described above. Above-referenced, [co-pending] commonly owned U.S. Patent 6,611,498 [Application No. _____ (D#11053)] also describes the class interactions when an icon is set on feature initialization. When a user presses a button on the main toolbar to launch a feature, e.g., NEMS, Rules, Provisioning, etc., the CMAAppView derived from the theMainToolbar class creates/activates the selected feature and initializes it. When the CMFeature is instantiated or started, it invokes a setIcon method to create a frame, the CMFeatureFrame, in which to run the selected feature.

Please amend the paragraph beginning on page 331, line 20, as follows.

Example MML and (SCP) host system status messaging supporting the system status display feature of the CMWS application is described in above-referenced, [co-pending] commonly owned U.S. Patent 6,611,498 [Application No. _____ (D#11053)]. When a message prompting user readiness is received by the host, the host sends system status messages on a regular interval. The length of the interval may vary anywhere from every second to every minute. The system status display messages are automatically sent from the SCP host, once the messages are turned on. The backend typically stores this type of message received from the SCP host until the client queries for it. A unique set of data is stored

for each user currently performing a system status display. Subsequent messages received from the SCP host typically overwrite previous data.

Please amend the paragraph beginning on page 333, line 17, as follows.

The CMWS component of the nMCI Interact system supports a branding functionality which allows users to open the call manager webstation application in a company specific context. Details including an example class diagram including classes used in branding process and, a scenario diagram showing an example of branding process for presenting a warning dialog with a company brand is found in above-referenced, [co-pending] commonly owned U.S. Patent 6,611,498 [Application No. _____ (D#11053)]. In this application, the CMBackPlane class is derived from the COBacklane class, which is an applet, which inherits all of the applet attributes and methods. This branding feature may be equally used in the other applications of the present invention when the Backplane class for that application is defined. The main URL for the branded application uses JavaScript, a client side scripting language, to render the html. The JavaScript, typically, directs the browser to retrieve a company brand. The browser then opens the application web page with the company web page specified in the query portion of the URL. Typically the application applet retrieves a company brand name by invoking a getParameter method (an applet method), and sets the brand name in the application specific globals class by invoking a setBrand method. This feature of the present invention enables resellers of enterprise telecommunication services to brand those services and features as their own when providing them to a third party user or customer. This facilitates the adoption and resale of unused enterprise capacity through a reseller market.

Please amend the paragraph beginning on page 335, line 2, as follows.

The CMWS component of the nMCI Interact system additionally provides an internationalization feature, supporting local languages for text displays. This optional feature allows a user to open the call manager application in a language as set by the user. Subsequent texts and phrases are rendered in the language chosen. Typically, the call manager webstation application is opened with a default language as set by the operating system. The user is also given an option to select a language other than the default. A call manager applet typically determines the locale set for the operating system and launches the appropriate language version by including the locale as a parameter. For example, the parameter with a name "locale" may have one of the following values: "en_US" for English US, "en_CA" for English-Canada, and "fr_CA" for French Canada. The applet uses this value to set the locale for the system string and phrase resources. A scenario diagram for setting the locale via the call manager applet is found in above-referenced, [co-pending] commonly owned U.S. Patent 6,611,498 [Application No. _____ (D#11053)]. As described in [co-pending] commonly owned U.S. Patent 6,611,498 [Application No.

____ (D#11053)] a CMResource class handles the general resources of character encoding, numeric formatting and date formatting.

Please amend the paragraph beginning on page 348, line 16, as follows.

With more particularity, the batch print option may allow customers to send a batch print job to be performed at the enterprise Intranet to the customers, e.g., via Federal Express, at a location specified by the customer. An example batch print data entry screen and pop-up confirmation screen can be found in commonly owned, [co-pending] commonly owned U.S. Patent 6,745,229 [Application No. ____ (D#11055)] entitled WEB BASED INTEGRATED CUSTOMER INTERFACE FOR INVOICE REPORTING, the contents and disclosure of which is incorporated by reference as if fully set forth herein.

Please amend the paragraph beginning on page 348, line 27, as follows.

Another feature provided by the ClientView system 1300 includes an accumulator function which allows customers to add up numerical figures, such as minutes and charges, by highlighting the numbers directly on the screen. Details regarding the accumulator function can be found in above-referenced, [co-pending] commonly owned U.S. Patent 6,745,229 [Application No. ____ (D#11055)].

Please amend the paragraph beginning on page 349, line 7, as follows.

The above-mentioned fax current document option offered by the online invoicing application includes an ability to fax to the customer, at a customer specified location, a current page, specified range of pages, or the entire document by making an appropriate selection in a fax dialog box such as described in above-referenced, [co-pending] commonly owned U.S. Patent 6,745,229 [Application No. ____ (D#11055)].

Please amend the paragraph beginning on page 354, line 10, as follows.

The information on documents for imaging and storing are typically received from the various networkMCI Interact's application services. A list of the various billing systems providing product feeds to online invoicing for document imaging is provided in above-referenced, [co-pending] commonly owned U.S. Patent 6,745,229 [Application No. ____ (D#11055)].

Please amend the paragraph beginning on page 389, line 18, as follows.

Further as shown in the DM7. 17 is a second RTM server 52 having its own connection to the public Internet via a TCP/IP connection 32. As described in [co-pending] commonly owned U.S. Patent

6,470,386 [Application No. _____, (D#11045)] this server provides real-time session management for subscribers of the network MCI Interact Real Time Monitoring system. An additional TCP/IP connection 48 links the RTM Web server 52 with the MCI Intranet Dispatcher server 26.

Please amend the paragraph beginning on page 393, line 23, as follows.

For example, in performing the verification, translation and communication functions, the Report Manager server, the Report Scheduler server and Inbox server proxies (Figure 10) each employ front end proxy C++ objects and components. For instance, a `utils.c` program and a C++ components library, is provided for implementing general functions/objects. Various C++ parser objects are invoked which are part of an object class used as a repository for the RM metadata and parses the string it receives. The class has a `build` member function which reads the string which contains the data to store. After a message is received, the parser object is created in the `RNDispatcher.c` object which is filed containing the business logic for handling metadata messages at the back-end. It uses the services of an `RMParse` class. Upon determining that the client has sent a valid message, the appropriate member function is invoked to service the request. Invocation occurs in `MCIRMServerSocket.C` when an incoming message is received and is determined not to be a Talarian message. `RMServerSocket.c` is a class implementing the message management feature in the Report Manager server. Public inheritance is from `MCIserverSocket` in order to create a specific instance of this object. This object is created in the main loop and is called when a message needs to be sent and received; a `Socket.c` class implementing client type sockets under Unix using, e.g., TCP/IP or TCP/UDP. `Socket.C` is inherited by `ClientSocket.C::Socket(theSocketType, thePortNum)` and `ServerSocket.C::Socket(theSocketType, thePortNum)` when `ClientSocket` or `ServerSocket` is created. A `ServerSocket.c` class implements client type sockets under Unix using either TCP/IP or TCP/UDP. `ServerSocket.C` is inherited by `RMServerSocket` when `RMServerSocket` is created. An `InboxParser.c` class used as a repository for the RN Metadata. The class' `"build"` member function reads the string which contains the data to store and the class parses the string it receives. After a message has been received, the `MCIInboxParser` object is created in `inboxutil.c` which is a file containing the functions which process the Inbox requests, i.e., Add, Delete, List, Fetch and Update. Additional objects/classes include `Environ.c` which provides access to a UNIX environment; `Process.c` which provides a mechanism to spawn slave processes in the UNIX environment; `Daemon.c` for enabling a process to become a daemon; `Exception.c` for exception handling in C++ programs, and, `RMlog.c` for facilitating RN logging. In addition custom ESQL code for RN/database interface is provided which includes the ESQL C interface (Informix) stored procedures for performing the ARD, DRD, DUR, URS, GRD, CRD, and GPL messages as described with reference to Appendix A in copending U.S. Patent Application No. _____ (D#11050). The functions call the stored procedures according to the

FROM MCI Technology Law Washington DC

(TUE) 5. 3' 05 9:01/ST. 8:45/NO. 4261173247 P 27

Docket No. COS97101

message, and the response is build inside the functions depending on the returned values of the stored procedures. A mainsql.c program provides the ESQ L C interface for messages from the report manager and report viewer.